VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

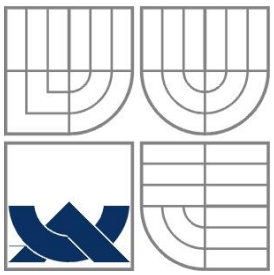
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MONITORING A ANALÝZA UŽIVATELŮ SYSTÉMEM DLP

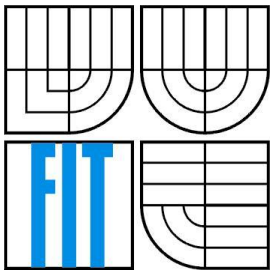
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
PETER LÁSZLÓ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MONITORING A ANALÝZA UŽIVATELŮ SYSTÉMEM DLP

MONITORING AND ANALYSIS OF USERS USING DLP SYSTEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PETER LÁSZLÓ
AUTHOR

VEDOUCÍ PRÁCE — Ing. MICHAL DROZD
SUPERVISOR

BRNO 2011

Abstrakt

Cílem této práce je seznámit čtenáře s tzv. Data Loss Prevention systémy, popsat jednotlivé části těchto systémů a jejich výhody a nevýhody. Dokument obsahuje potřebné znalosti o filtru IRP zpráv a o samotných IRP zprávách. Pomocí uvedených znalostí je možné navrhnout klientskou aplikaci, která je, na principu Data Loss Prevention, schopna chránit citlivá data od různých forem útoku. Vytvořená aplikace je realizována jako mini-filtr IRP zpráv, který, prostřednictvím monitorování a filtrování IRP zpráv, umožňuje sledovat práci s vybranými soubory. Dokument dále obsahuje podrobný návrh, implementaci, testování a zhodnocení vytvořeného programu.

Abstract

The purpose of this work is presenting Data Loss Prevention systems, describing each element of it and writing down advantages and disadvantages of these systems. This document contains required knowledge about filter drivers and I/O request packets. According to this knowledge it is possible to design a user application, which based on the principles of Data Loss Prevention systems is able to prevent sensitive data from leaks. Created application is a mini-filter driver, which allows tracking operations of marked files with monitoring and filtering IRP packets. This document also contains detailed desing, implementation, testing and reviewing of created program.

Klíčová slova

Data Loss Prevention, mini-filter, ovladač, IRP, jádro

Keywords

Data Loss Prevention, mini-filter, driver, IRP, kernel

Citace

Peter László: Monitoring a analýza uživatelů systémem DLP, bakalářská práce, Brno, FIT VUT v Brně, 2011

Monitoring a analýza uživatelů systémem DLP

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Drozda. Další informace mi poskytl Ľuboš Hnaníček. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Peter László
17.05.2011

Poděkování

V tejto sekcii by som chcel poďakovať môjmu vedúcemu, Ing. Michalovi Drozdovi, za všetky rady, trpezlivosť a čas, ktorý strávil pri konzultáciách. Ďalej by som chcel poďakovať Ľubošovi Hnaníčkovi zo spoločnosti Trustport a.s. za odborné rady pri návrhu aplikácie a spoločnosti AEC spol. s.r.o. za poskytovanie technológii a informácie k nim.

Obsah

1 Úvod

S rozšírením počítačov v osobných a pracovných oblastiach života narástla aj kybernetická kriminalita a spolu s ňou aj potreba ochrany dát. Existuje široká paleta bezpečnostných programov, ako firewally, intrusion detection systémy, intrusion prevention systémy, antivírusové produkty atď., ale ani jeden z nich nie je dostatočne zameraný na ochranu citlivých dát. Preto môže dôjsť k internému úniku spôsobeným nevhodným chovaním užívateľa alebo útočníka.

Práve na monitorovanie citlivých dát existujú špecializované Data Loss Prevention (ďalej len DLP) systémy. Pomocou bezpečnostných pravidiel nám umožnia sledovať prácu s vybranými dátami a zakázať nežiadané operácie, ako poslanie dát cez verejné siete alebo otváranie duševného vlastníctva neautorizovaným osobám. Tieto systémy so zameraním na ochranu citlivých dát pracujú na troch vrstvách. Prvá je *Data in motion*, kde je rekonštruovaný dátový tok na sieti a analyzovaný obsah. Táto činnosť umožní blokovanie úniku cez siete. Druhú časť systému tvorí *Data in use*, ktorá slúži na monitorovanie práce užívateľa so sledovanými súbormi. Ako príklad si môžeme uviesť zakázanie premiestnenia dát na USB nosiče. Poslednou časťou systému je *Data at rest*. Táto časť pomocou špeciálneho programu *Crawler*[1] skúma dátové sklady a hľadá citlivé dáta, ktoré zatiaľ nie sú katalogizované. Nevýhoda DLP systémov spočíva práve v komplexnosti, čo spôsobí spomalenie systému a stávajú sa tak nevhodným pre osobné použitie.

Cieľom práce je objasniť úlohu Data Loss Prevention systémov a vytvoriť užívateľskú aplikáciu na princípe DLP systémov, ktorá pomocou monitorovania operácií nad vybranými súbormi bude schopná úspešne zabrániť úniku dát. Aplikácia bude zameraná na zabezpečenie dát, bez ohľadu na typ útoku. Interným cieľom je, aby výsledný program bol ľahko použiteľný, znemožnil možné druhy útokov a zároveň nespomalil rýchlosť operačného systému v značnej miere.

Práca je štruktúrovaná podľa obsahu do viacerých logických jednotiek. V druhej kapitole sú podrobne rozpísané DLP systémy a je odôvodnené nutnosť vzniku týchto systémov. Sú uvedené výhody a zároveň aj nevýhody DLP systémov a ako príklad je uvedené existujúce komerčné riešenie. V tretej kapitole sú rozoberané systémové správanie užívateľov a sú kategorizované formy úniku dát. Ďalej v tejto časti sú popísané možnosti a používané metódy na analýzu obsahu súborov. Ďalšia kapitola obsahuje návrh aplikácie a potrebné teoretické znalosti o ovládačoch systému Windows. Popisuje možné miesta a techniky monitorovania súborov. Definuje jednotlivé časti, z ktorých sa program skladá. V piatej kapitole sú rozobrané zaujímavejšie implementačné detaily aplikácie. Táto časť práce tiež obsahuje popis vývojových prostredí, v ktorých bola aplikácia vytvorená. Šiesta kapitola je miestom testov. V tejto sekcii je skúšaná odolnosť programu proti rôznym útokom. Je tiež testovaná správanie a rýchlosť systému pri väčšom počte monitorovaných súborov. Je popísaná v akej miere spomalí aplikácia rýchlosť operačného systému pri sledovaní tisícich súborov. V závere sú v krátkosti zhrnuté dosiahnuté výsledky. Sú diskutované možnosti reálneho uplatnenia vytvoreného systému. Posledná kapitola tiež obsahuje výhľad do budúcna a možné vylepšenie aplikácie.

[1] Program **Crawler** je detailne rozpísaná v kapitole 2.2.

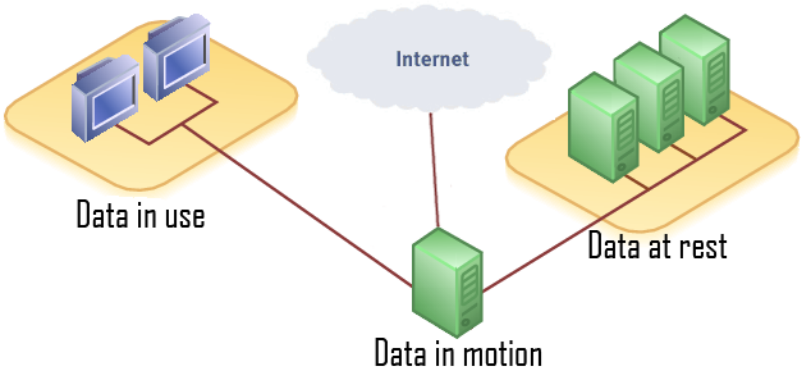
2 Data Loss Prevention systémy

Na ochranu počítačových systémov proti útokom sú používané rôzne miesta, techniky a bezpečnostné programy. Podľa miesta ochrany dát môžeme deliť aplikácie na sieťové a koncové. Na siete sú najčastejšie používané systémy Firewall, Intrusion Detection (IDS) a Intrusion Prevention (IPS) systémy. Na koncových zariadeniach sú väčšinou používané rôzne anti-malwareové produkty, ako anti-vírové a anti-spywareové aplikácie. Cieľom týchto programov je filtrovanie, detekcia a zabránenie podozrivých činností. Spoločnou nevýhodou týchto aplikácií je, že pracujú podľa vopred definovaných detekčných profilov či signatúr. Technika signatúr spočíva v skúmaní súborov a obsahu pamäti. Tento obsah je porovnaný s prvkami databáz, ktoré obsahujú signatúry známych počítačových vírov. Detekčné profily používajú rôzne algoritmy na odhalenie škodlivých programov na základe spoločných čŕt nebezpečných aplikácií [1]. Zmienené mechanizmy znamenajú neschopnosť okamžitej reakcie na zatiaľ neznáme útoky. Pomocou týchto aplikácií je tiež nemožné zabrániť internému úniku dát. Dôvodom vzniku DLP systémov je riešiť tieto nedostatky z pohľadu úniku dát.

Cieľom DLP systémov je identifikovať, monitorovať a predovšetkým chrániť citlivé dáta. Prácu DLP môžeme rozdeliť na 3 kľúčové časti (viď obrázok č. 1):

- Data at rest, ktorá má za úlohu lokalizovanie a katalogizovanie dáta v dátových skladoch (viď kapitolu 2.2);
- Data in motion, ktorá monitoruje a kontroluje dátový tok na sieti (viď kapitolu 2.3);
- Data in use, ktorou úloha je monitorovanie a kontrolovanie práce na koncových zariadeniach (viď kapitolu 2.4).

Kapitoly 2.2, 2.3 a 2.4 sú parafrázované z [2]. Obsah kapitoly 2.6 je čerpaná z [3].



Obrázok 1: Data Loss Prevention systémy a 3 kľúčové časti.

2.1 Ľudský faktor v bezpečnosti

Ľudský faktor v IT bezpečnosti IT bol vždy známou hrozbou. Pod týmto pojmom nemyslíme na škodlivú činnosť hackerov, či na externé útoky. Práve naopak, ľudský faktor predstavuje nevhodné správanie zamestnancov, či už úmyselne alebo nevedome.

Dôkaz vážnosti interného úniku dát ukazuje aj [4], kde v roku 2009 pokladali zamestnancov za najväčšiu a v roku 2010 za druhú najväčšiu hrozbu úniku dát. Podľa [5] je v bezpečnosti dát ľudský faktor podcenený, čo vyplýva z faktu, že 90% bezpečnostných kontrol je zameraných na externé útoky, pričom 70% informačných zločinov je vykonaných pracovníkmi firmy. Štandardné bezpečnostné produkty, ako napr. firewally, sú zamerané na zabezpečenie lokálnych sietí proti externým hrozbám. Proti interným útokom nás ale nechránia, čo predstavuje kritický bod zraniteľnosti pre skladované informácie.

Systém DLP je zameraný na ochranu citlivých dát, bez ohľadu na pôvod útoku, čo v drvivej väčšine znemožňuje aj interný únik dát. Princíp práce a detailnejší popis jednotlivých častí DLP je uvedený v nasledujúcich kapitolách.

2.2 Data at rest

Kľúčová činnosť DLP systémov je identifikácia miesta uloženia citlivých dát. Výhodou tejto funkcie je umožnenie aplikovania politiky, bez ohľadu na miesta uloženia dát, na spôsob zdieľania a na spôsob používania. Napríklad je možné definovať bezpečnostnú politiku, ktorá vyžaduje šifrovanú formu čísiel kreditných kariet cez emaily, zakáže zdieľanie týchto údajov cez protokoly *HTTP*[2] či *HTTPS*[2] a umožní uloženie výhradne na počítačoch účtovníckeho tímu.

Na identifikáciu miesta uloženia citlivých dát sú používané tzv. *crawleri*. *Crawleri* sú programy, ktoré prehľadávajú zariadenia a dátové skladiská a identifikujú citlivé dáta. Majú tri hlavné zložky. Prvá je *Endpoint Discovery*, ktorá prehľadáva koncové zariadenia. Druhou zložkou je *Storage Discovery*, ktorá ma za úlohu skenovať dátové skladisko. Posledná zložka je *Server Discovery*, čo je špecifické hľadanie dát na emailových serveroch či v databáze. Používajú sa 3 techniky na objavenie citlivého obsahu:

- *Remote Scanning*: vytvára sa pripojenie medzi crawlerom a kontrolovaným zariadením a skenovanie sa vykonáva diaľkovo.
- *Agent-Based Scanning*: na kontrolovaný systém je nainštalovaný *crawler* a skenovanie je uskutočnené lokálne.
- *Memory-Resident Agent Scanning*: do pamäti je inštalovaný *crawler*, ktorý po kontrole ukončí vlastný beh, bez uchovania dát na lokálnom systéme.

2.3 Data in motion

Na zachytávanie a analýzu sieťového toku sú využívané špecifické zariadenia. Hlavným prvkom je pasívny *network monitor*. Tento prvok je typicky nasadený do alebo v blízkosti defaultnej brány. *Network monitor* v reálnom čase zachytáva packety, rekonštruuje dátový tok a analyzuje obsah.

Druhým dôležitým prvkom je kontrola elektronických správ. Využíva sa tzv. *Mail Transport Agent* (ďalej len MTA), ktorý je pridaný ako fyzický prvok do cesty emailov. Nevýhoda tohto prístupu je, že neumožní kontrolovanie interných emailov.

Aby bola ochrana úspešná, je potrebné tok dát blokovať. Nikto nechce vedieť o úniku citlivých informácií bez schopnosti to prerušiť. Blokovať a prepustiť správne toky a analyzovať v reálnom čase však nie je jednoduché. Realizácia je možná buď cez most, cez proxy alebo cez TCP poisoning. Most

[2] **HTTP** a **HTTPS** sú protokoly aplikačnej vrstvy modelu TCP/IP, ktoré sú popísané v [24]

je zariadenie pripojené do siete, cez ktorý je obsah toku dát kontrolovaný. Keď sa odhalí podozrivý obsah, tak most zruší pripojenie pre príslušnú *session*. Proxy je protokol špecifického zariadenia, ktorý zaradí tok do fronty pred jeho odoslaním, čo umožní hlbšiu analýzu. Tieto techniky zahŕňajú protokoly ako *HTTP* či *FTP*[3], ale existujú DLP systémy, ktoré vedia analyzovať i *SSL*[3].

Posledná metóda sa volá TCP[4] Poisoning. TCP Poisoning monitoruje tok dát na sieti a keď odhalí nevhodný obsah, tak injektuje TCP reset packet na prerušenie pripojenia. Tento spôsob prerušenia funguje na každý TCP protokol, nie je však efektívny. Sú protokoly, ktoré sa budú aj naďalej pokúšať o posielanie. Napríklad štandardný emailový server bude opakovať pokus každých 15 minút po dobu troch dní. Druhou nevýhodou tejto techniky je to, čo je aj nevýhodou zariadení typu most. Vzhľadom na to, že tok nie je zaradený do fronty, môže sa nastať situácia, keď v čase odhalenia neautorizovaného obsahu je už prerušenie neskoré.

2.4 Data in use

DLP systémy v tejto časti monitorujú pohyb dát vyplývajúci z akcií užívateľov na koncových zariadeniach, ako kopírovanie dát na USB disk, posielanie dát cez *peer-to-peer* aplikácie, či *copy-paste* citlivého obsahu. Existujúce riešenia sa značne líšia funkcionalitou, ale je možné zdôrazniť tri kľúčové činnosti:

- Monitorovanie sieťového zásobníka. Táto technika umožní vynútiť sieťové pravidlá bez sieťových zariadení.
- Monitorovanie v rámci jadra systému. Práca v kernel leveli nám umožní monitorovanie činnosti užívateľa, ako napr. otvorenie citlivých informácií.
- Monitorovanie súborového systému nám umožní zakázať operácie, ako premiestnenie dôležitých informácií na nešifrované USB zariadenie.

2.5 Komerčné riešenie

Prostredníctvom spoločnosti AEC spol. s.r.o. (www.aec.cz) som mal možnosť vyskúšať komerčné riešenie DLP systému od firmách Websense (www.websense.com) a McAfee (www.mcafee.com).

V tejto kapitole predvedieme ukážku z akých častí sa skladá vyskúšaný DLP systém od firmy Websense. Testovaný produkt má názov *Websense Data Security Suite*. Verzia testovaného produktu bola 7.5. Tento balíček aplikácií zahrňuje štyri integrované moduly. Informácie o tomto softwaru som čerpal z [6].

Prvý modul je *Websense Data Monitor*, ktorý monitoruje sieť a dátové toky na nej. Poskytuje nám monitorovať webové a mailové protokoly.

Druhý modul je *Websense Data Protect*, ktorý zahrňuje aj modul *Websense Data Monitor*. *Websense Data Protector* zaisťuje uplatňovanie a dodržiavanie automatických kontrolných mechanizmov podľa politiky, ktoré slúžia k blokovaniu dát, ukladaniu dát do karantény, informovaniu užívateľa na porušenie pravidiel.

[3] **FTP** a **SSL** sú protokoly aplikačnej vrstvy modelu TCP/IP, o ktorých je možné nájsť detailný popis v [24]
[4] **Transmission Control Protocol** (**TCP**) je protokol transportnej vrstvy modelu TCP/IP, ktorý je popísaný v [24]

Websense Data Endpoint je tretí modul systému. Monitoruje a zaisťuje uplatňovanie automatických kontrolných mechanizmov na dáta používané aplikáciami v koncových bodoch. Prehľadáva miestne dáta a umožní klasifikovať dôverné informácie.

Posledný modul je *Websense Data Discover*. Tento modul vyhľadáva a klasifikuje dôverné dáta uložené v dátových skladoch.

2.6 Výhody a nevýhody

Ako každý bezpečnostný program aj systém DLP má ako primárny cieľ chrániť systém od útokov. Prvotný cieľ a najväčšia výhoda týchto riešení je ochrana citlivých dát a duševného vlastníctva pred neoprávneným použitím. Nezanedbateľným prínosom je aj optimalizácia diskového priestoru a šírky pásma siete. DLP systémy identifikujú stagnujúce súbory a streamované videá, ktoré konzumujú veľké množstvo zdrojov, ako aj miesto na serveroch. Čistenie zatuchnutých súborov a zabránenie s obchodnou činnosťou nesúvisiacich video streamov môže vo významnej miere redukovať požiadavky na zálohovanie a skladovanie dát. Ďalšou výhodou týchto systémov je detekcia škodlivých aplikácií, ktoré pokúšajú prenášať citlivé dáta cez internet.

Napriek vyššie uvedeným výhodám DLP systémy majú aj svoje nevýhody, ako zaťaženie systému a sieti. Systém stále kontroluje dáta s ktorými užívateľ pracuje, čo môže mať vplyv na rýchlosť zariadenia a aj práca *crawlerov* (viď kapitolu 2.2) v istej miere spomaľuje systém. So stálym kontrolovaním dátového toku na sieti môže byť rýchlosť komunikácie spomalená. Zmienené nevýhody by pritom nemali v značnej miere byť na úkor práce so systémom. Dnešné DLP systémy sú navrhnuté pre spoločnosti s viacerými počítačmi. Je predpokladané, že rôzne časti Data Loss Prevention systémov budú aplikované na rôznych zariadeniach. To na príklad znamená oddelenie centrálnej správy od časti Data at use. Stanú sa tak nevhodným na osobné použitie, lebo rýchlosť dnešných počítačov s priemerným výkonom nie je dostačujúca na spustenie všetkých častí DLP systému na jednom zariadení.

Pre úplnosť tejto problematiky je potrebné sa zmieniť o tom, že ani jeden bezpečnostný program nezabezpečuje systém v plnej miere a ani DLP systémy nie sú výnimkou. Ako významné limitácie si môžeme pripomenúť:

- Šifrovanie. DLP systém je schopný kontrolovať šifrované dáta, ktoré môže najprv dešifrovať. Aby DLP agenti, sieťové zariadenia a *crawleri* mali možnosť dešifrovať dáta, potrebujú poznať šifrovaciu metódu a šifrovací kľúč. Keď užívatelia sú schopní používať osobné šifrovacie balíčky, kde heslá nie sú spravované centrálnou správou, tak DLP systémy nemajú možnosť analyzovať dáta v reálnom čase.
- Grafika. DLP systém nemôže inteligentne interpretovať grafické súbory, ako napr. skenované dáta alebo grafické formáty súborov, ani geometrické a matematické modely súčiastok.
- Mobilné zariadenie. S rozšírením inteligentných mobilných zariadení vznikajú nové komunikačné kanály, ktoré nie je možné monitorovať, či kontrolovať. Ako príklad si môžeme uviesť krátke textové správy (SMS). Významnejšou hrozbou slúžia zariadenia ktoré majú možnosť využiť Wi-Fi alebo dokonca sa môžu stať prístupovými bodmi sami. Schopnosť dnešných mobilných telefónov vytvárať fotografie a nahrávať audio záznamy predstavujú ďalšie nekontrolovateľné toky dát systému.

Ako záver tejto kapitoly si môžeme vyhlásiť, že s využitím DLP systémov sa riziko straty citlivých dát výrazne znižuje, je však pre úspešnú ochranu odporúčané aplikovať aj iné ochranné programy, ako firewall či antivírusové aplikácie.

3 Analýza systémového správania užívateľov

Aby sme boli schopní vytvoriť úspešnú bezpečnostnú aplikáciu, potrebujeme poznať užívateľa a jeho činnosť, formy útokov vedúcich k strate alebo krádeži dát a metódy, s ktorými môžeme úspešne zabrániť týmto činnostiam.

3.1 Formy útokov

Formy útokov a spôsob straty dát môžeme rozdeliť do štyroch hlavných kategórií, ktoré sú nechcené zverejňovanie informácií, náhodná strata dát, nesprávne odstránenie dát a úmyselná krádež informácií (viď obrázok č. 2).

Prvú kategóriu tvorí nechcené zverejňovanie informácií. Tieto nehody sa najčastejšie nastanú na webových stránkach, či cez e-maily. Napriek maximálnemu úsiliu organizácií užívatelia naďalej používajú e-maily na distribuovanie citlivých dát. Jedna z najľahších miest IT útokov sú práve mailové účty. Keď užívateľ spravil chybu a dopustil sa indiskrécie, je najviac pravdepodobné, že informácie nájdeme práve v e-mailoch. Užívatelia sa často domnievajú, že ich firemné mailové účty sú sledované (v skutočnosti sa to stáva len zriedkavo), alebo ich zamestnávatelia majú prístup k nim a používajú radšej účty, o ktorých si myslia že sú súkromné (napr. služby ako Hotmail, Yahoo Mail, GMail[5]). Niektoré organizácie vyriešili tento problém zakázaním prístupu na web-maily z ich sietí. Toto riešenie má aj iné výhody ako napr. úsporu šírky pásma siete. Hoci úniky cez e-poštu sú bežné, majú aspoň obmedzený rozsah. Existujú nehody, kde niekto stlačí tlačítko "reply all" namiesto tlačítka "reply" a privátnu správu dostane väčšia skupina ľudí. Táto situácia nemusí znamenať v každom prípade katastrofu, za určitých okolností však môže viesť k nepríjemnostiam. Únik dát cez webové stránky je potenciálne oveľa viac škodlivý. Historicky najčastejšou príčinou úniku sú programátorské chyby, ktoré umožnia návštevníkom stránok prístup k informáciám ku ktorým nemajú práva [7].

Druhá kategória je náhodná strata dát. Možno najlepším príkladom sú straty či krádeže počítačových zariadení. Do tejto kategórie zahŕňame aj fyzickú krádež laptopov, pretože v prevažnej

Obrázok 2: Možné formy úniku citlivých dát.

[5] **Hotmail**, **Yahoo Mail** a **GMail** sú bezplatné e-mailové služby, ktoré sú prístupné formou webového rozhrania.

väčšine prípadov sú ukradnuté pre ich vlastnú hodnotu a nie kvôli hodnote dát uložených na nich. Notebooky však nie sú jediným problémom. USB kľúče sú prítomné už v každej sfére života. V dnešnej dobe nie je nezvyčajné, že niekto vlastní viacero USB zariadení. Tieto nástroje môžu nosiť obrovské množstvo dát a kvôli lacnej cene ľudia nevenujú dostatočnú pozornosť strate týchto zariadení [7].

Tretiu skupinu útokov tvorí nesprávne odstránenie informácií, keď dôjde k likvidácii počítačov či iných pamäťových médií. Jednoduché vymazanie súborov alebo formátovanie pevných diskov neodstráni dáta uložené na ňom. Počítače sú určené na ukladanie dát rýchlo a spoľahlivo. Naopak pri odstraňovaní už nie sú tak efektívne. Pre väčšinu moderných počítačových zariadení to znamená, že odstránené dáta je možné obnoviť [7].

Poslednú kategóriu tvorí úmyselná krádež informácií. Väčšina týchto krádeží dát je vykonaná vlastným zamestnancom spoločnosti. Najčastejšie sa táto činnosť prejavuje po zrušení zamestnaneckého vzťahu. Hlavnými nosičmi odcudzených dát sú USB zariadenia. Ako to už bolo spomenuté, tieto nástroje sú schopné niesť veľké množstvo dát, ktoré môžu byť kopírované rýchlo a neviditeľne. Druhým problémom kategórie zostáva hackovanie, aj keď tejto činnosti je menej ako v minulosti. Hackeri zistili, že je výhodnejšie uskutočniť rozsiahle podvody, ako sú phishing[6] útoky, ktoré sú viac založené na zneužití dôvery ako na technickej znalosti. V posledných rokoch boli spoločnosti často napadnuté prostredníctvom trójskych koní[7]. Často boli aplikované pomocou zamestnancov, ktorým boli zaslané pripojením do e-mailov. Medzi hackerov je dobre známym pravidlom, že človek je najmenej bezpečným prvkom počítačového systému [7].

Táto práca bude zameraná na obranu proti prvej a poslednej kategórií straty dát, čo je nechcené zverejňovanie informácií prostredníctvom world wide webu, ako náhodné zverejňovanie citlivých dát na webových portáloch, a úmyselná krádež dát. Dôvodom výberu týchto typov úniku dát je, že majú spoločné rysy z technického hľadiska. Prvá spoločná vlastnosť je virtuálne odcudzenie informácií, bez ohľadu na to, či bolo plánované alebo nie. Druhý dôvod, ktorý tieto kategórie spája, je fakt, že dáta boli odcudzené z firemných systémov.

3.2 Bezpečnostné udalosti a politiky

Informačná bezpečnosť znamená chrániť informácie a informačné systémy pred neoprávneným prístupom, použitím, sprístupňovaním, prerušením, zmenám či zničením [10]. Aby sme tieto požiadavky boli schopní aplikovať pomocou DLP systémov, potrebujeme presne definovať čo chceme chrániť a pred čím.

Na tento účel DLP systémy používajú bezpečnostné politiky, ktoré definujú, ktorí užívatelia aké operácie môžu či nemôžu vykonať s určitými dátami. Príkladom bezpečnostnej politiky môže poslúžiť už zmienený prípad, pomocou ktorého vyžadujeme šifrovanú formu čísiel kreditných kariet cez emaily, zakážeme zdieľanie týchto údajov cez protokoly *HTTP* či *HTTPS* a umožníme uloženie týchto dát len na počítačoch účtovníckeho tímu.

Porušenie bezpečnostných pravidiel označujeme ako bezpečnostné udalosti. Presnou definíciou je udalosť označujúca porušenie alebo zlyhanie bezpečnostných politík [11]. Príkladom

[6] **Phishing** označuje činnosť, pri ktorej sa podvodník snaží vylákať od používateľov rôzne heslá. Jedna z metód phishingu je založenie webovej stránky, ktorá vyzerá ako presná kópia už existujúcej dôveryhodnej stránky. Meno a heslo zadané do phishingovej stránky, sa odošlú podvodníkovi, ktorý ich môže zneužiť [8].

[7] **Trójsky kôň** je program, ktorý vykonáva deštruktívnu činnosť, pričom sa tvári ako užitočný [9].

bezpečnostnej udalosti v prípade vyššie uvedenej bezpečnostnej politiky môže byť posielanie čísiel kreditných kariet ako obsah email správy bez šifrovania, alebo zdieľanie čísiel kreditných kariet cez protokol *HTTP*.

Úspešnosť zabránenia úniku citlivých dát pomocou DLP systémov závisí plne od kvality špecifikovania bezpečnostných politík. Nekvalitný návrh bezpečnostných politík môže spôsobiť neschopnosť detekcie úniku dát či vysoký počet generovaných false positives[8].

3.3 Analýza obsahu

Veľkú rolu u vynucovaní bezpečnostných politík v DLP systémoch hrá analýza obsahu. Je potrebné analyzovať každú činnosť, ktorá sa odohráva v systéme a na sieti. Pre obyčajný text nie je táto úloha náročná. Situácia sa značne komplikuje u binárnych súborov. Systémy DLP to riešia pomocou metódy tzv. *file cracking*. *File cracking* je technológia používaná k čítaniu súboru, aj v prípade, keď jeho obsah je viacero úrovňový. Ako príklad si môžme uviesť Excel tabuľky, ktoré sú vložené do Word dokumentu, ktorý je komprimovaný. DLP tento súbor najprv musí dekomprimovať, načítať Word dokument, nájsť tabuľky a až po tom to analyzovať. Po sprístupnení obsahu sa používajú rôzne techniky na analýzu. Techniky používané u väčšiny DLP systémov sú popísané v nasledujúcej časti kapitoly spolu s výhodami a nevýhodami. Hlbšie znalosti o týchto technikách som získal z [2].

Prvá technika je analýza s regulárnymi výrazmi. Pomocou nich je možné v obsahu hľadať podľa špecifických pravidiel, ako napr. 16 číselné reťazce kreditných kariet. Ako prvý filter sú

```
^(?:(?<Visa>4\d{3})|(?<Mastercard>5[1-5]\d{2})|(?
<Discover>6011)|(?<DinersClub>(?:3[68]\d{2})|(?
:30[0-5]\d))|(?<AmericanExpress>3[47]\d{2}))([
-]?)(?(DinersClub)(?:\d{6}\1\d{4})|(?(AmericanExpr
ess)(?:\d{6}\1\d{5})|(?:\d{4}\1\d{4}\1\d{4})))$
```

Obrázok 3: Regulárny výraz čísiel kreditných kariet.

ideálne na detekciu ľahko identifikovateľných dát. Postup kontroly je rýchly, ale sú náchylné generovať vysoký počet false positive. Príkladom regulárneho výrazu je na obrázku č. 3, pomocou ktorého je možné vyfiltrovať čísla kreditných kariet.

Druhá technika sa volá *Exact Data Matching*, ktorá využíva databázu odtlačkov, kde hľadá presné zhody dát. Výhodou tejto metódy je možnosť hľadania štruktúrovaných dát a nízky počet false positive, hľadanie v databáze však spomalí systém.

Tretia technika v rade sa volá *Exact File Matching*. Metóda *Exact File Matching* berie hash súboru a monitoruje súbory s rovnakým odtlačkom. Umožní monitorovať binárne súbory, kde textová analýza nie je možná. Nevýhodou tejto techniky je, že aj malá zmena v súbore spôsobí zmenu vo vygenerovanej hash hodnote.

Partial Document Matching je štvrtou technikou, ktorá hľadá kompletné alebo čiastkové zhody v obsahu. Táto metóda je založená na technike *Cyclical hashing*, ktorá berie cyklicky jednotlivé časti obsahu, z čoho generuje hash hodnotu. Túto činnosť opakuje, až kým nedosiahne koniec dokumentu. Výhodou techniky je schopnosť chrániť neštruktúrované dáta. Nevýhodou je obmedzenie výkonnosti systému.

[8] **False positive** znamená falošné označenie užitočnej činnosti za škodlivé.

Piata technika je štatistická analýza, ktorá využíva strojové učenie sa a iné štatistické techniky na analýzu a chránenie obsahu. Je schopný upozorniť na obsah, ktorý sa podobá na chránené dáta. Nevýhodou techniky je, že počet false positive u štatistickej analýze môže byť značná.

Predposledná technika je encyklopedická analýza, ktorá používa slovníky, pravidlá a iné analýzy na nájdenie citlivého obsahu. Hľadá v texte kľúčové frázy, kontroluje počet a pozíciu slov. Nevýhoda metódy je vo vysokom počte generovaných false postivite.

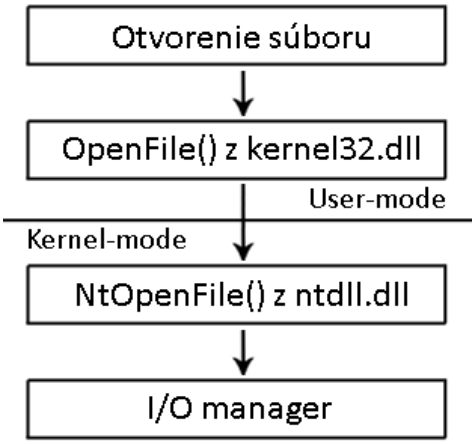
Posledná metóda je kategorizácia. Kategorizácia pracuje s pravidlami a so slovníkmi na nájdenie často sa opakujúcich citlivých dáta. Konfiguruje sa jednoducho, ale je to metóda použiteľná len pre jednoducho kategorizované obsahy.

4 Návrh aplikácie

Cieľom práce je vytvoriť bezpečnostnú aplikáciu na operačný systém Windows, ktorá na princípe DLP systémov bude chrániť vybrané citlivé dáta. Na rozdiel od DLP systémov bude aplikácia navrhnutá tak, aby bola primárne použiteľná pre užívateľov osobných počítačov. Medzi hlavné účely patrí úspešná ochrana bez ohľadu na spôsobu útoku. Program by mal úspešne zabrániť virtuálnym (napr. cez sieť) aj fyzickým (napr. odcudzenie pri neprítomnosti vlastníka u počítača) útokom. Ďalej je cieľom, aby vytvorená aplikácia nespomalila v značnej miere rýchlosť operačného systému, ani pri väčšom množstve monitorovaných súborov. V tejto časti práce pod pojmom operačný systém sa rozumie operačný systém Microsoft Windows.

4.1 Hooking

Na zabezpečenie dát je potrebné monitorovať prácu s nimi. Kontrolovať prácu so súborom je možné pomocou techniky *hooking*. *Hooking* predstavuje spôsob pri ktorom rozšírime alebo zmeníme správanie operačného systému pomocou zachytávania API funkcií, či správ [12]. Aby sme boli schopní hookovať funkcie operačného systému, potrebujeme vedieť čo sa odohráva pri práci so súborom.



Obrázok 4: Činnosť operačného systému pri otváranie súboru.

Ako príklad si uvedieme zjednodušenú činnosť operačného systému pri otváraní textového súboru pomocou funkcie *OpenFile()*, ktorá je znázornená na obrázku č. 4. Udalosť sa začína s otvorením súboru pomocou WinAPI funkcie *OpenFile()* z knižnice *kernel32.dll*. Z *OpenFile()* je volaná funkcia *NtOpenFile()* z jadra operačného systému, ktorá je uložená v dynamicky linkovanej knižnici *ntdll.dll*. V ďalšom kroku je pomocou I/O manageru generovaná príslušná IRP (I/O Request Packet) správa.

Sledovať a modifikovať činnosť operačného systému je možné v každom uvedenom kroku, avšak som sa rozhodol monitorovať IRP správy systému. Dôvodom rozhodnutia je fakt, že nie je povinné aby aplikácia využila vopred implementované funkcie ako *OpenFile()* či *NtOpenFile()*. Tým pádom je možné aby aplikácia využila iné funkcie s rovnakou funkcionalitou, čo by spôsobilo

vyhnutie sa vytvoreným hookom. Pred samotným návrhom aplikácie je ale nutné sa oboznámiť s I/O managerom operačného systému Windows a so štruktúrou IRP správ.

4.2 I/O Manager

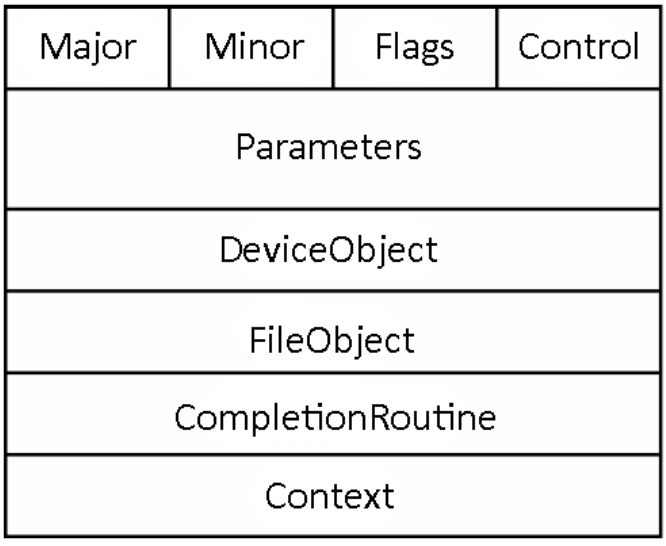
I/O[9] manager je hlavnou častou vstupných a výstupných operácií v jadre systému, lebo definuje model a poradie, v ktorom sú vstupné a výstupné žiadosti doručené k ovládačom. I/O systém je riadený packetmi. Väčšina vstupných a výstupných žiadostí je reprezentovaná pomocou IRP správ, ktoré sú poslané medzi komponentami I/O systému. Úlohou I/O manageru je vytvorenie príslušnej IRP správy a doručenie k vybraným ovládačom. Taktiež má za úlohu po ukončení operácií IRP správu likvidovať. Oproti tomu ovládač dostane IRP packet od I/O manageru, vykoná operáciu definovanú v správe a vráti IRP packet I/O managerovi [13].

4.3 IRP

Input/Output Request Packet je dátová štruktúra, ktorá obsahuje všetky informácie o I/O žiadosti. Keď vlákno zavolá vstupnú alebo výstupnú službu operačného systému, I/O manager vytvorí IRP správu, ktorá reprezentuje postup operácie v I/O systéme [13].

IRP správa sa skladá z dvoch častí, z pevnej hlavičky a z hromady. V pevnej časti správy sú informácie ako typ a veľkosť dotazu. Kedykoľvek je vytvorená IRP správa, je spolu s ňou vytvorená aj hromada *IO_STACK_LOCATION* štruktúr. V hromade je uložený jeden stack location pre každý ovládač, ktorý bude pracovať s IRP správou [14]. Z pohľadu monitorovania udalostí nie je pre nás dôležitá pevná hlavička IRP správy. Potrebné informácie o I/O operácii sú v štruktúre *IO_STACK_LOCATION*, ktorá sa skladá z nasledujúcich častí (viď obrázok č. 5):

- *Major funkcia* je hlavným kódom funkcie súvisiacej s IRP správou. Môže mať hodnotu ako



Obrázok 5: Časti štruktúry IO_STACK_LOCATION.

[9] Skratka **I/O** znamená input/output, v preklade vstup/výstup

MJ_READ, podľa ktorej je priradená k funkcii z tabuľky MajorFunction v ovládači.

- *Minor funkcia* slúži na doplňujúce identifikácie IRP správ. Minor typ kódu sa využíva len u niektorých hlavných IRP správach ako *IRP_MJ_READ* či *IRP_MJ_WRITE*.
- *Parameter* je únia sub-štruktúr. Sú to špecifické štruktúry pre každý typ žiadostí.
- *Device Object* je adresa zariadenia, ktorá zodpovedá aktuálnej položke v zásobníku.
- *File Object* je adresa jadrového súboru, ku ktorému je IRP správa smerovaná.
- *Completion Routine* je adresa ukončujúcej rutiny.
- *Context* je ľubovoľná kontextová hodnota, ktorá je predaná ako argument ukončujúcej rutiny [14].

Existujú rôzne typy IRP major packetov, ktoré sú vymenované a popísané na stránke [15]. Pre monitorovanie práce so súborom je však dostačujúce sledovať *MJ_CREATE* a *MJ_SET_INFORMATION*. Eventuality generovaní týchto správ sú popísané v tabuľke č. 1.

Dôvodom výberu je, že pomocou typu *MJ_CREATE* sme schopní monitorovať každý pokus o otvorenie, kým *MJ_SET_INFORMATION* nám umožní sledovanie vybraných zmien súboru.

Major funkcia	Prípad generovania
CREATE	Ak je nový súbor alebo adresár vytvorený, alebo keď existujúci súbor, zariadenie či adresár je otvorený
SET_INFORMATION	Ak sú zmenené vlastnosti objektu

Tabuľka 1: Typ a popis monitorovaných IRP správ.

4.4 Device a Filter Driver

IRP správy sme schopní monitorovať pomocou device driveru. *Device driver* je komponenta, ktorý operačný systém používa na poskytovanie I/O služieb. Rozdeľujeme tri typy *device driverov*:

- *Function drivers*, ktoré riadia jednotlivé zariadenia;
- *Filter drivers*, ktoré filtrujú I/O žiadosti;
- *Bus drivers*, ktoré majú za úlohu riadiť bus kontrolóry, adaptéry a mosty [16].

Na uskutočnenie našich cieľov je možné použiť *Filter driver*. Konkrétne *file system filter driver*, ktorý je umiestnený nad *file system driverom*. *File system filter driver* umožní monitorovať, doplniť či modifikovať všetky vstupné a výstupné žiadosti v súborovom systéme [13].

4.5 Filter Manager

Filter manager je *file system filter driver*, ktorý vytvoril Microsoft na zjednodušenie vytvárania third-party *filter driverov* a vyrieši veľa problémov existujúceho *legacy filter driver* modelu. *Filter driver* vyvinutá pre Filter Manager je nazvaná *minifilter driver* [17]. Vytvorenie *mini-filter driverov* má svoje výhody i nevýhody. Nevýhodou je, že filter manager sa nachádza implicitne v systémoch Windows 2000 SP4 a v novších modeloch. Výhodou implementovania *mini-filter driverov* je viacero:

- jednoduchší vývoj;
- dynamické nahranie, uvoľnenie, pripojenie a odpojenie driveru;

- pripojenie na definované miesto v zásobníku filtrov;
- podpora nerekurzívnych I/O správ, aby I/O správy generované *mini-filterom* boli viditeľné len pre driveri nachádzajúce sa pod *mini-filterom*;
- filtrovanie výlučne tých operácií o ktorých má *mini-filter* záujem.

Inštalácia *mini-filter driverov* je umožnená pomocou *INF* súborov, kde sú definované ktoré inštancie bude driver podporovať [18].

4.6 Návrh Personal DLP

Po predchádzajúcich znalostiach sme schopný navrhnúť aplikáciu na základnú ochranu proti zneužitiu citlivých dát. Vytvorený systém bude obsahovať dve hlavné časti a dva pomocné súbory. Hlavné časti budú *file system mini-filter driver* (ďalej len driver) a užívateľská aplikácia. Pomocné súbory budú používané na ukladanie informácií. Prvý súbor má názov log súbor a bude obsahovať všetky zaznamenané udalosti a detaily o nich. Druhým pomocným súborom je konfiguračný súbor, ktoré bude miestom na uloženie bezpečnostných pravidiel.

Detailnejší popis jednotlivých častí je obsahom nasledujúcich podkapitol. Navrhnutý systém budeme nazývať Personal DLP, kvôli jednému z vytýčených cieľov, čo je návrh na možné použitie na osobné počítače.

4.6.1 Driver

Hlavnou úlohou driveru je monitorovanie a filtrovanie IRP správ, konkrétne správy *IRP_MJ_CREATE* a *IRP_MJ_SET_INFORMATION*.

Pozorovaním *IRP_MJ_CREATE* packetov sme schopný vyfiltrovať pokusy o otvorenie súborov. Pred otvorením súboru, driver obdrží I/O request packet, ktorý obsahuje všetky potrebné informácie o operácii, ako meno súboru, ktoré chce užívateľ otvoriť či prístupové práva užívateľa. Podľa mena súboru sa driver rozhodne, či súbor je monitorovaný alebo nie. V prípade ak súbor patrí do pozorovanej časti, tak po kontrole príznakov pokus zakáže či povolí.

Kontrolovanie *IRP_MJ_SET_INFORMATION* packetov je potrebné kvôli užívateľských operácií ako premiestnenie či premenovanie súboru. Podrobnejší popis a rozklad tejto problematiky je v kapitole 4.6.3. Driver bude mať tiež za úlohu ukladať zaznamenané udalosti do log súboru.

4.6.2 Spôsoby monitorovania

Driver bude rozdeľovať prácu so súborom do dvoch skupín: *Read* a W*rite*. Rozpoznanie skupiny sa bude diať podľa parametrov správy *IRP_MJ_CREATE*, konkrétne podľa bitovej masky *DesiredAccess*. *DesiredAccess* špecifikuje typ prístupu, ktorý vyžaduje volajúci proces. Do skupiny *Read* budú patriť hodnoty *FILE_READ_DATA*, *FILE_READ_ATTRIBUTES*, *FILE_READ_EA*, *READ_CONTROL*. Druhá skupina zahrňuje príznaky *FILE_WRITE_DATA*, *FILE_WRITE_ATTRIBUTES*, *FILE_WRITE_EA*, *FILE_APPEND_DATA*, *WRITE_DAC*, *WRITE_OWNER* a *DELETE*. Význam jednotlivých príznakov je rozpísaný v tabuľke č. 2.

Pre jednotlivé skupiny bude mať užívateľ možnosť nastaviť 3 reakcie na príslušné hodnoty. Prvá hodnota ja nazvaná *No Log*. Pomocou *No Log* sme schopní nastaviť aby udalosť nebola monitorovaná. V tomto prípade driver operáciu povolí. Druha reakcia sa volá *Log*, kde práca so

súborom je monitorovaná a operácia povolená. Poslednou možnosťou je *Deny*. Pri tejto hodnote driver generuje log a operáciu zakáže. Skupina *Read* a *Write* u monitorovaných súboroch môže mať rôzne hodnoty.

Príznak	Význam	Skupina
FILE_READ_DATA	Dáta môžu byť načítané zo súboru	Read
FILE_READ_ATTRIBUTES	Atribúty súboru je možné čitať	Read
FILE_READ_EA	Rozšírené atribúty súboru je možné čítať	Read
READ_CONTROL	Access Control List (ACL) a informácie o vlastníctva je možné čítať	Read
FILE_WRITE_DATA	Dáta je možné písať do súboru	Write
FILE_WRITE_ATTRIBUTES	Atribúty súboru je možné zmeniť	Write
FILE_WRITE_EA	Rozšírené atribúty je možné zmeniť	Write
FILE_APPEND_DATA	Dáta je možné pridať do súboru	Write
WRITE_DAC	Discretionary ACL súboru je možné zmeniť	Write
WRITE_OWNER	Informácie o vlastníctva je možné zmeniť	Write
DELETE	Súbor môže byť odstránený	Write

Tabuľka 2: Príznaky bitmasku DesiredAccess a triedenie príznakov do skupín.

V prípade ak si užívateľ zvolí pre skupinu *Read* hodnotu *Deny*, driver automaticky nastaví hodnotu *Deny* aj pre skupinu *Write*. Dôvod tejto vlastnosti je zvyšovanie efektivity. Keď operačný systém nemá možnosť čítať zo súboru, tak predpokladáme, že nechceme dovoliť ani písanie do súboru.

4.6.3 Premenovanie, premiestnenie a kopírovanie

Pre dosiahnutie cieľa je potrebné aby vytvorený driver bol schopný identifikovať operácie ako premenovanie či premiestnenie monitorovaného súboru a upraviť konfiguračný súbor podľa zmeny. Premenovanie či premiestnenie súboru v operačnom systéme Windows sú identické. Tieto operácie môžeme rozobrať na nasledujúce IRP správy:

1. Otvorenie zdrojového súboru (*IRP_MJ_CREATE*);
2. Nastavenie zmeny názvu súboru (*IRP_MJ_SET_INFORMATION*);
3. Zatvorenie súboru (*IRP_MJ_CREATE*).

Pomocou analýzy *IRP_MJ_SET_INFORMATION* packetov, ktoré obsahujú počiatočné aj nové meno súboru, je možné tieto operácie odhaliť a pridať bezpečnostnú politiku do konfiguračného súboru s modifikovaným názvom súboru.

Na kopírovanie súboru táto operácia už nie je možná. Duplikovanie súboru je uskutočnené s nasledujúcimi IRP správami:

1. Otvorenie zdrojového súboru (*IRP_MJ_CREATE*);
2. Otvorenie cieľového súboru (*IRP_MJ_CREATE*);
3. Načítanie zdrojového súboru do pamäti (*IRP_MJ_READ*);
4. Zápis do cieľového súboru z pamäti (*IRP_MJ_WRITE*), keď je v pamäti ďalší obsah na kopírovanie tak skok na 3. bod, ak nie tak skok na 5. bod;

5. Nastavenie dátum poslednej modifikácie cieľového súboru (*IRP_MJ_SET_INFORMATION*);
6. Zatvorenie zdrojového súboru (*IRP_MJ_CLOSE*);
7. Zatvorenie cieľového súboru (*IRP_MJ_CLOSE*).

Na identifikovanie operácie kopírovania súboru by driver potreboval rozpoznať, že *IRP_MJ_CREATE* správy z bodu 1. a 2. patria k sebe. To však s plnou istotou nie je možné zaručiť, lebo tieto *IRP_MJ_CREATE* správy obsahujú len jedno meno súboru (cieľové či zdrojové) a nie je isté, že druhá *IRP_MJ_CREATE* správa bude nasledovať prvú.

Druhou možnosťou je monitorovať *IRP_MJ_SET_INFORMATION* packet z bodu 5. Takáto správa je ale generovaná aj pri bežnej modifikácii akéhokoľvek súboru, čo by spôsobil v tomto prípade veľký počet false positives.

Kvôli uvedeným nedostatkom monitorovania kopírovania súborov pomocou IRP správ nebude driver schopný rozpoznať a správne reagovať na túto operáciu.

4.6.4 Užívateľská aplikácia

Druhou častou projektu je užívateľská aplikácia, ktorá má za úlohu vytvárať komunikačný kanál medzi užívateľom a driverom. Pomocou nej je schopný užívateľ spravovať konfiguračný súbor (pridať či odstrániť bezpečnostnú politiku) a prečítať si obsah log súboru.

Užívateľská aplikácia parsuje konfiguračný súbor a vytvorený zoznam politík posiela driverovi. Pri načítaní obsahu konfiguračného súboru je nutné overovať existenciu monitorovaných súborov a neexistujúce súbory odstrániť z konfiguračného súboru. Príčinou výskytu neaktuálnych záznamov v zozname je spôsob práce drivera. Driver pri premiestnení či premenovaní monitorovaných súborov pridá bezpečnostnú politiku s modifikovaným menom súboru, neaktuálnu politiku už ale neodstráni. Preto užívateľská aplikácia pred komunikáciou s driverom aktualizuje zoznam, aby driver nebol zaťažený zastaralými prvkami.

Užívateľská aplikácia umožňuje dočasné vypnutie pravidiel, aby užívateľ nemusel pred prácou s monitorovanými súbormi odstrániť a po práci pridať tú istú bezpečnostnú politiku. Prístup k väčšine úkonov užívateľskej aplikácie je chránený heslom.

4.6.5 MS-DOS Device name a Internal NT Device name

V operačnom systéme Windows na užívateľskej úrovni sú používané tzv. *MS-DOS Device names* (ďalej len dosové mená), kde zariadenie je označené pomocou *Drive letter*. Ako príklad si môžeme uviesť cestu a meno súboru *„c:\Users\User\file.txt“*. Z uvedeného mena zariadenie je označené pomocou *„c:“*.

Na úrovni jadra však nie je používaný tento typ označenia. Používa sa tzv. *Internal NT Device names* (ďalej len interné mená), v ktorom je označený typ a meno zariadenia. Predchádzajúci príklad s textovým súborom by mal formu *„\Device\HarddiskVolume1\Users\User\file.txt“*. Dosové mená súborov nazývame symbolickými odkazmi interných mien [19].

Užívateľská aplikácia bude vyžadovať, aby užívateľ zadal príslušný súbor pomocou dosových mien, keďže na užívateľskej úrovni sa nepoužívajú interné označenia a bolo by pre užívateľa náročné ručné konvertovanie ciest. Avšak po zadaní a po otestovaní existencie bude cesta súboru konvertovaná na internú formu.

V konfiguračnom súbore a v log súbore budú tiež používané súbory s internou formou. Dôvodom je, že do konfiguračného súboru má možnosť písať ako aj ovládač (pri operácií premenovania a premiestnenia súborov), takisto aj užívateľská aplikácia (pri pridaní nového súboru), a kvôli konzistencie konfiguračného súboru je konvertovanie nevyhnutné.

4.6.6 Konfiguračný súbor

Konfiguračný súbor bude miesto na uloženie bezpečnostných politík. Modifikovať tento súbor bude ako aj užívateľská aplikácia, tak aj driver. Obecný tvar jednej položky v konfiguračnom súbore je:

Read;Write;\Device\Harddisk\file.ext,

kde *Read* a *Write* reprezentujú reakciu na skupinu operácií popísané v kapitole 4.6.2. Možné hodnoty *Read* a *Write* sú *NoLog, Log* alebo *Deny*, podľa požadovanej reakcie, ktoré sú popísané v kapitole 4.6.2. Poslednú časť záznamu tvorí cesta a meno monitorovaného súboru. Príklad jedného záznamu v konfiguračnom súbore je:

Log;Deny;\Device\HarddiskVolume1\Program Files\dir\picture.bmp.

Obsah konfiguračného súboru bude šifrovaný, aby nebolo možné zobraziť jednotlivé záznamy bez užívateľskej aplikácie.

4.6.7 Log file

Obsah log súboru budú tvoriť zaznamenané udalosti. Pridanie záznamov bude úlohou driveru. Rola užívateľskej aplikácie bude sprístupnenie log súboru užívateľovi. Aplikácia umožní čítanie jednotlivých záznamov, či zmazanie obsahu súboru. Log súbor bude tak ako aj konfiguračný súbor, šifrovaný kvôli bezpečnosti. Uložené budú dva typy záznamov.

Prvý typ je zaznamenanie či zakázanie udalostí, ktorý bude mať tvar:

Time: YYYY.MM.DD HH:MM:SS.MS process operation reaction,\Device\Harddisk\file.ext.

Prvá časť, *YYYY.MM.DD HH:MM:SS.MS*, reprezentuje čas a dátum výskytu udalosti. *Process* je nahradený menom procesu, ktorý operáciu vyvolá. V sekcii *operation* je uložená skupina operácie. Má jednu z hodnôt *Read* alebo *Write*. *Reaction* predstavuje reakciu na vyvolanú operáciu, môže mať hodnoty *Logged* či *Denied*. Posledná časť slúži na uloženie mena súboru. Príkladom uloženia zakázaných udalosti do záznamu je:

Time: 2011.4.27 20:22:43.833 notepad.exe Write Denied, \Device\HarddiskVolume2\file.txt.

Pomocou druhého typu záznamu bude uložené premenovanie a premiestnenie súborov. Obecný typ tohto záznamu bude:

Time: YYYY.MM.DD HH:MM:SS.MS Move/Rename reaction from:
\Device\Harddisk\old.ext to: \Device\Harddisk\new.ext.

Záznam po výskytu času a kľúčových slov *Move/Rename* obsahuje meno súboru, z ktorého bol súbor premenovaný (*\Device\Harddisk\old.ext*) a meno modifikovaného súboru (*\Device\Harddisk\new.ext*). *Reaction* je reakcia ovládača na operáciu premenovania či premiestnenia súboru. Možné hodnoty *reaction* sú *Logged a Denied*. Príkladom zaznamenaného premenovania sledovaného súboru je:

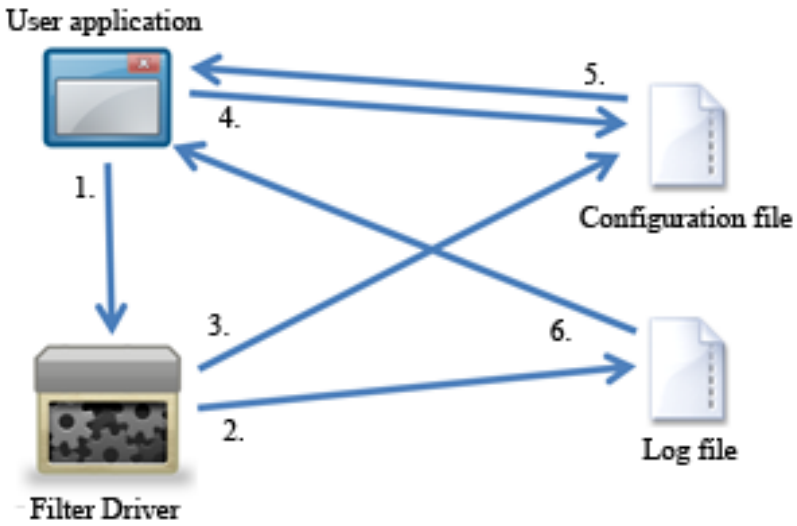
Time: 2011.5.10 17:22:12.121 Move/Rename Logged from:
\Device\HarddiskVolume1\dir\file.txt to:
\Device\HarddiskVolume1\new_file.txt.

4.6.8 Dátový tok a spôsob komunikácie

Po návrhu všetkých častí aplikácie Personal DLP sme schopní zhrnúť možné dátové toky a spôsoby komunikácie, ktoré sú znázornené a očíslované na obrázku č. 6. Dátový tok je možné rozdeliť na dve skupiny.

Prvú skupinu tvoria dátové toky, ktoré sú zahájené driverom. Na obrázku č. 6 sú reprezentované šípkami s číslom 2 a 3. Šípka označená číslom 2 popisuje vytvorenie logovacích záznamov driveru a uloženie ich do log súboru. Šípka s číslom 3 predstavuje činnosť driveru pri premenovaní či premiestnení monitorovaného súboru, pri ktorom driver doplní konfiguračný súbor záznamom, ktorý obsahuje aktualizované meno a miesto citlivých informácií.

Druhú skupinu dátových tokov tvoria tie, ktoré sú zahájené užívateľskou aplikáciou, na



Obrázok 6: Dátový tok v Personal DLP.

obrázku č. 6 označené číslami 1, 4, 5 a 6. Číslo 1 reprezentuje komunikáciu s driverom, ktorá zahrňuje viacero operácií, ako poslanie zoznamu súborov či dočasné vypnutie a zapnutie vybraných pravidiel. Šípka označená číslom 6 je čítanie obsahu log súboru. Šípka s číslom 4 a 5 reprezentujú

prácu užívateľskej aplikácie s konfiguračným súborom. S číslom 4 označená šípka označuje vykonané zmeny v konfiguračnom súbore, ako pridanie či odstránenie bezpečnostných pravidiel. Posledná šípka s číslom 5 reprezentuje činnosť užívateľskej aplikácie, pri ktorej sa číta obsah konfiguračného súboru.

5 Implementácia

Projekt z hľadiska implementácie je možné rozdeliť do dvoch logických skupín.

Prvou je implementovanie ovládača. Ovládač je vytvorený v jazyku C a preložený pomocou príkazu *build*. Program *build* je súčasťou Windows Driver Kitu (ďalej len WDK). WDK je plne integrované prostredie na vývoj ovládačov, a je dostupný na stránke [20]. Na preklad bola v práci použitý WDK s verziou 7.1 (7600.16385.1).

Druhou častou je vytvorenie užívateľskej aplikácie. Táto časť bola vytvorená ako projekt vo vývojárskom prostredí Microsoft Visual Studo 2010 (ďalej len VS2010) v jazyku C++. Verzia prostredia, v ktorom bola aplikácia vytvorená jej 10.0.30319.1.

5.1 Ovládač

Základná kostra ovládača Personal DLP bola vytvorená zo vzorového riešenia *mini-filter driveru PassThrough*, ktorá je súčasťou WDK.

Driver sa skladá z viacerých funkcií ktoré sú nevyhnutné pre úspešné monitorovanie IRP správ. Prvá funkcia je *DriverEntry()*, ktorú volá Filter manager. V tejto funkcii je inicializovaný ovládač a je vytvorený komunikačný port pomocou *FltCreateCommunicationPort()*. Obsahom funkcie je tiež spustenie filtrovania IRP správ.

Druhá funkcia je *Unload()*, ktorá je volaná pri vypnutí driveru. Slúži na uvoľnenie zoznamov, ukončenie komunikačného kanálu a na odregistrovanie ovládača.

Funkcia *PreOperationSetInfo()* je volaná, ak ovládač obdrží IRP správu typu *MJ_SET_INFORMATION*. Obsah funkcie je detailnejšie rozpísaný v kapitole 5.1.2. *PreOperationCreate()* je volaná pri IRP správe *MJ_CREATE*. Popis činnosti funkcie je obsahom kapitoly 5.1.1.

Funkcia *ClientConnect()* je volaná pri pripojení užívateľskej aplikácie na komunikačný kanál, *ClientDisconnect()* pri jej odpojení. Poslednou hlavnou funkciou je *ClientMessage()* ktorá spracuje doručené správy od užívateľskej aplikácie. V Personal DLP existuje 6 typov správ, ktoré sú uvedené v tabuľke č. 3 s popisom reakcie ovládača na príslušné správy pri ich obdržaní.

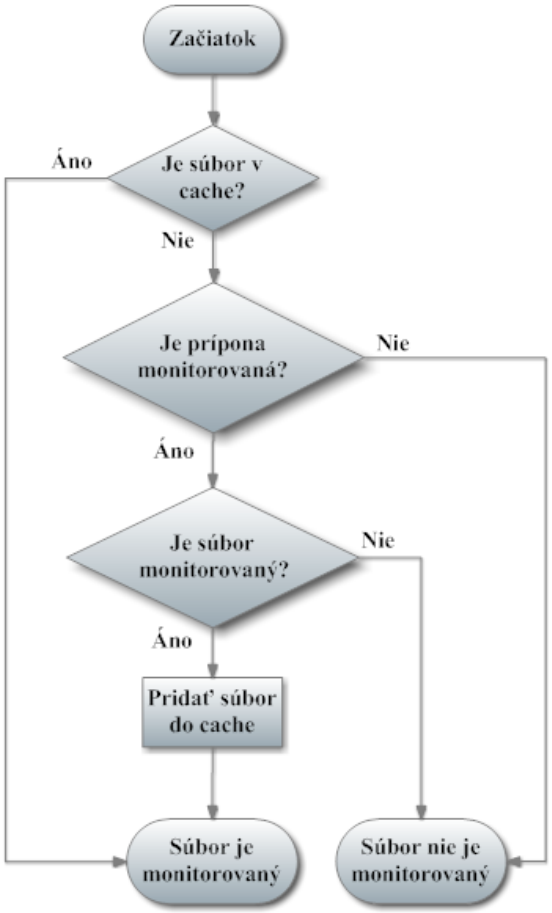
Typ správy	Popis reakcie ovládaču pri obdržaní
MSG_DEL_LISTS	Uvoľnenie pamäti alokované pre cache a zoznamov
MSG_RULES_ON	Zapnutie všetkých pravidiel
MSG_RULE_ON	Aktivovanie vybraného pravidla
MSG_RULE_OFF	Deaktivovanie vybraného pravidla
MSG_EXT	Pridanie záznamu správy do zoznamu prípon
MSG_FILE	Pridanie záznamu správy do zoznamu súborov

Tabuľka 3: Typ a popis komunikačných správ v Personal DLP.

I/O Manager pošle IRP správu typu *MJ_CREATE*, keď súbor či adresár je vytvorený alebo, keď existujúci súbor, adresár či zariadenie je otvorené. Takýchto operácií je obvykle veľa a nie sú vykonané len užívateľom. Aj operačný systém vykonáva otváranie súborov, ktoré sú pre užívateľa

transparentné. Jedným z cieľov pri návrhu aplikácií bolo, aby výsledný program nespomalil rýchlosť operačného systému pri behu.

Pre tento účel sú implementované v ovládači 3 rôzne zoznamy, pomocou ktorých sa filtrujú IRP správy. Prvým zoznamom je tzv. cache, kde je uložených 10 posledne použitých monitorovaných súborov. Druhý zoznam, zoznam prípon, slúži na uloženie monitorovaných prípon. Tretí zoznam je zoznam monitorovaných súborov. Vývojový diagram filtrovania IRP správ je zobrazený na obrázku č. 7. Pri filtrovania IRP správ ovládač najprv prehľadá cache. Ak cache obsahuje súbor, ktorý je uložený v IRP správe, tak je monitorovaný. V opačnom prípade ovládač prekontroluje zoznam prípon. Ak prípona súboru sa nenachádza v zozname, tak je súbor vyhodnotený ako nemonitorovaný a IRP správa je vrátená I/O managerovi. Keď prípona súboru bola nájdená v zozname monitorovaných prípon, tak je prehľadaný posledný zoznam v ktorom sú uložené monitorované súbory. Keď ovládač nájde súbor v zozname, tak ho pridá do cache a vyhodnotí ho ako monitorovaný. V opačnom prípade je IRP správa vrátená I/O managerovi. V situácii keď IRP správa



Obrázok 7: Vývojový diagram viacúrovňového filtrovania IRP správ.

bola vyhodnotená ako monitorovaná, tak sú kontrolované jeho príznaky ovládačom. Podľa bezpečnostnej politiky je operácia buď povolená alebo status IRP správy je zmenený na *STATUS_ACCESS_DENIED* a vrátený I/O managerovi. Pomocou tohto statusu ovládač operáciu zakáže, čo užívateľ v operačnom systéme registruje ako chybové hlásenie o nedostatku práv alebo o zamietnutí prístupu.

Preklad ovládača je možné vykonať pomocou príkazu *build* v prostredí Build, ktorý je súčasťou WDK. Nainštalovať vytvorený *sys* súbor je možné pomocou *inf* súboru, ktorý je na priloženom CD nosiči. Spustenie nainštalovaného ovládača sa uskutočňuje pomocou Filter manageru cez príkazový riadok s príkazom *fltmc load pdlp*.

5.1.1 Funkcia PreOperationCreate

Úlohou funkcie *PreOperationCreate* je správne reagovať na operácie so sledovanými dátami. Funkcia je volaná pri obdržaní IRP správy typu *MJ_CREATE*.

Na začiatku funkcie ovládač vykoná test, po ktorom je zistené či súbor patrí medzi sledované. V prípade ak súbor nie je monitorovaný ovládač vráti IRP správu I/O managerovi bez zmien. Ak súbor patrí medzi sledované, z príslušnej bezpečnostnej politiky je zistená nastavená reakcia na skupinu operácií *Read* a *Write*. V prípade ak súbor je obsahom viacerých bezpečnostných politík je vždy aplikovaná politika s najmenším indexom. V závislosti od bezpečnostnej politiky sú nastavené logované a zakázané príznaky. To napríklad znamená, pri nastavenej reakcie *Log* na skupinu operácií *Write*, pridanie príznakov skupiny *Write* z tabuľky 2 do logovaných príznakov. Po týchto nastaveniach sú testované príznaky súboru v IRP správe. V závislosti od toho, či nastavené príznaky sú zakázané alebo logované je pri zhode IRP správa vrátená I/O managerovi buď so štatusom *STATUS_ACCESS_DENIED* alebo bez zmien.

5.1.2 Funkcia PreOperationSetInfo

Funkcia *PreOperationSetInfo* je volaná v prípade, ak ovládač obdrží IRP správu od I/O managera s major kódom *MJ_SET_INFORMATION*.

Cieľom funkcie je identifikovať premiestnenie či premenovanie monitorovaného súboru. Tieto operácie z pohľadu operačného systému sú identické, preto v tejto kapitole pod operáciou premenovanie bude myslená aj operácia premiestnenie. *IRP_MJ_SET_INFORMATION* správa, ktorá reprezentuje operáciu premenovanie súboru či adresára, má triedu nastavenú na *FileRenameInformation*.

Ovládač Personal DLP vráti bez zmien I/O managerovi *IRP_MJ_SET_INFORMATION* správy, ktoré majú inú triedu ako *FileRenameInformation*. Po identifikovaní monitorovaných operácií je vo funkcii testované či meno súboru v IRP správe je monitorované alebo nie. Ak súbor patrí medzi sledované, tak je z IRP správy získané aktualizované meno. Ovládač skontroluje, či nové meno monitorovaného súboru obsahuje znak „.“, ktorý reprezentuje oddeľovač mena a prípony súboru. V prípade neprítomnosti oddeľovača je premenovanie súboru zakázané. V opačnom prípade pred povolením operácie ovládač vykoná bezpečnostné opatrenia. V prvom rade je súbor pridaný do dynamického zoznamu súborov a prípon. Ako nasledujúci krok je súbor s aktualizovaným menom pridaný do konfiguračného súboru. Po týchto činnostiach je IRP správa vrátená I/O managerovi.

V log súbore sú zaznamenané všetky pokusy o premenovanie bez ohľadu na úspešnosť.

5.2 Užívateľská aplikácia

Na umožnenie komunikácií užívateľovi s driverom a na spravovanie konfiguračného a log súboru bola vytvorená konzolová aplikácia v jazyku C++. Aby bolo možné z užívateľskej aplikácie sa pripojiť na vytvorený mini-filter driver sú potrebné pomocné funkcie z hlavičkového súboru

fltUser.h. Na používanie tohto súboru vo VS2010 bolo potrebné vykonať zopár úprav v nastaveniach projektu. V konfiguračnom nastavení projektu bolo pridané do *Include directories* a *Library directories* miesto uloženia WDK. Do nastavení *Linker* do časti *Additional Dependencies* bola pridaná knižnica *fltLib.lib*.

Aplikácia je rozdelená do viacerých súborov. V *user.cpp* sú spracované parametre a zavolané potrebné funkcie. V *params.cpp* sú implementované možné parametre. *Error.cpp* obsahuje chybové hlásenie aplikácie. V *functions.cpp* sú uložené pomocné funkcie, ako šifrovanie či spracovanie hesla. Hlavičkový súbor *structures.h* je miestom pre konštantné premenné, štruktúr a definícií.

Užívateľská aplikácia má 11 implementovaných spôsobov spustenia cez príkazový riadok. Prvým je spustenie programu bez parametrov. V tomto prípade program načíta z konfiguračného súboru všetky bezpečnostné politiky. Z načítaných dát vytvorí zoznam súborov a zoznam monitorovaných prípon. Potom sa pripojí na komunikačný port ovládača a pošle správu typu *MSG_DEL_LISTS*, pomocou ktorého vyprázdni cache a zoznamy ovládača. Na konci pošle obsah zoznamu monitorovaných prípon a súborov. Druhým spôsobom je pridanie bezpečnostnej politiky do konfiguračného súboru. Aplikácia to umožní pomocou parametru *-add*. V tomto prípade sú žiadané od užívateľa 3 ďalšie parametre. Prvý je názov súboru s celou cestou. Posledné dva parametre sú reakcie na skupinu operácií *Read* a *Write*, ktoré môžu mať hodnotu *nolog*, *log* a *deny*. Ďalší spôsob programu je možnosť odstránenia prvkov z konfiguračného súboru. Vykonanie týchto operácií je možné pomocou parametru *-del číslo*, kde číslo je index nepotrebnej bezpečnostnej politiky. Nasledujúce 3 spôsoby spustenia slúžia na zapnutie a vypnutie vybraných bezpečnostných politík. Pomocou parametru *-ruleoff index* máme možnosť na vypnutie vybranej politiky. Na aktivovanie vypnutej politiky slúži parameter *-ruleon* spolu s indexom bezpečnostnej politiky. Pomocou parametra *-ruleson* má užívateľ možnosť aktivovať všetky vypnuté pravidlá. Na vypísanie obsahu konfiguračného súboru spolu s indexmi bezpečnostných pravidiel slúži parameter *-echoconf*. Výpis záznamov z log súboru je možné pomocou *-echolog*. Na vyprázdnenie konfiguračného súboru a log súboru slúžia parametre *-clearconf* a *-clearlog*. Posledný implementovaný argument je *-help* pomocou ktorého je vypísaný pomocná stránka na štandardný výstup.

Na vykonanie operácií vyžaduje užívateľská aplikácia z bezpečnostných dôvodov zadanie hesla. Jediný parameter, ktorý je možné vykonať bez zadanie hesla je *-help*. Predvolené heslo v Personal DLP je „pass“.

Na komunikáciu s driverom sú používané preddefinované funkcie z *fltUser.h*. Pred poslaním samotnej správy je potrebné sa pomocou *FilterConnectCommunicationPort()* pripojiť na komunikačný port, ktorý vytvoril ovládač. Po úspešnom pripojení sme schopný pomocou *FilterSendMessage()* poslať potrebné dáta.

V kapitole 4.6.5 už bola spomenutá potreba konvertovania mien súborov. Na konvertovanie dosových mien na internú formu je vytvorená funkcia *convertDosNameToNTName()*, ktorá dostane ako parameter reťazec obsahujúci cestu a meno súboru. V reťazci vyhľadá znak diskovej jednotky a pomocou funkcie *QueryDosDevice()* konvertuje meno zariadenia na interné. Na konci vráti cestu a meno súboru v internej forme. Za účelom otestovania existencii súborov v konfiguračnom súbore je potrebné taktiež konvertovať z interných mien na dosové. Na tento účel bola vytvorená funkcia *convertNTNameToDosName()*.

Personal DLP používa konštantné mená a miesta uloženia log súboru a konfiguračného súboru. Oba súbory sú uložené v adresári *C:\Windows*. Dôvodom voľby umiestnenia pomocných súborov je,

že užívateľ bez administrátorských práv nemá možnosť pracovať so súbormi v tomto adresári. Meno log súboru je *PDLPlog.txt*, kým konfiguračný súbor je uložený pod menom *PDLPconf.txt*.

Ako už bolo spomenuté obsah konfiguračného a log súboru sú šifrované pomocou symetrickej šifry. Šifra používaná v programe pracuje na jednoduchom posúvacom princípe. Ku každému znaku textu je pridaná hodnota jedného znaku kľúča. Znaky z kľúča sú vyberané lineárne a cyklicky od prvého znaku. Predvolený kľúč šifry je „pdlpsecretkey“.

Preklad programu v VS2010 je možné uskutočniť v ponuke Build/Build Solution. Po preklade je možné spustiť program *user.exe*, ktorý je vytvorený v adresári *Debug*, s príslušnými parametrami.

6 Testovanie Personal DLP

Táto kapitola je venovaná testovaniu Personal DLP. Testy sú rozdelené do 3 skupín a vyskúšané pod užívateľskými právami na operačných systémoch Windows XP SP2 a Windows 7.

Prvú skupinu testov tvoria základné bezpečnostné testy, kde sú popísané reakcie na jednoduché užívateľské operácie. Druhá skupina obsahuje pokročilé bezpečnostné testy, ako útok cez sieť. Poslednú skupinu testov tvoria testy spomalenia systému pri použitý Personal DLP.

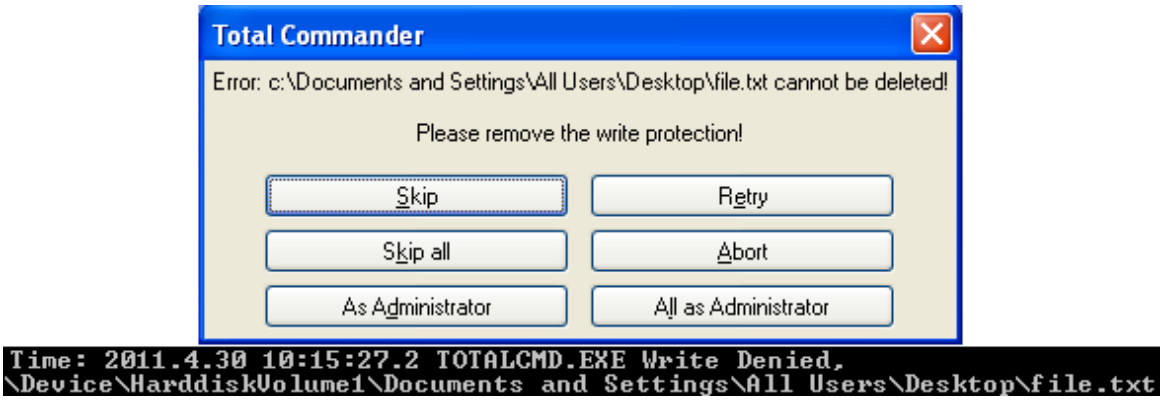
6.1 Základné bezpečnostné testy

V tejto časti práce sú popísané reakcie Personal DLP na základné operácie v operačnom systéme Windows.

V prvom teste je skúšaná úspešnosť monitorovania čítacích operácií. Pomocou parametra *-add* nastavíme aby aplikácii Personal DLP zákaz skupiny operácií *Read*. V teste je používaný textový súbor. Pri pokuse o otváranie súboru v operačnom systéme sa zobrazí okno s textom „Access is denied“. Po overení obsahu log súboru vidíme, že aplikácia vytvorila záznam s parametrami *Read Denied*.

V druhom základnom teste sú kontrolované reakcie vytvoreného driveru v prípade ak si užívateľ nastaví skupinu operácií *Write* na reakciu *Deny*. V teste je skupina operácií *Read* nastavená na *No Log*. Užívateľ v tomto prípade má právo na otváranie a preskúmanie obsahu monitorovaného súboru. Zmenu a uloženie vybraného súboru nám operačný systém už neumožní. Po neúspešnom uložení však operačný systém väčšinou ponúkne možnosť na uloženie súboru pod iným menom. Táto operácia je kopírovanie súboru. Monitorovanie tejto činnosti aktuálna verzia Personal DLP neumožní, z dôvodov popísaných v kapitole 4.6.3. Výsledok duplikovania súboru už nebude monitorovaný. Tento nedostatok systému môžeme považovať za najväčšiu chybu programu a je popísaná s možnými riešeniami v kapitole 6.3.

Pomocou nastavenia skupiny *Write* na hodnotu *Deny* ovládač Personal DLP úspešne zabráni vykonaniu operácie so súborom ako premiestnenie, premenovanie či odstránenie. Pri každom pokuse týchto operácií nám vyskočí varovanie o nedostatku práva užívateľa. Pri každom neúspešnom pokuse je log súbor obohatený ďalším záznamom, ktorá má parametre *Write Denied*. Príklad záznamu a varovanie operačného systému o nedostatku práv sú zobrazené na obrázku č. 8.



Obrázok 8: Varovanie operačného systému o nedostatku práv na odstránenie súboru a príklad vytvoreného záznamu v log súbore.

V nasledujúcom teste je vyskúšaná reakcia programu Personal DLP na premenovanie či premiestnenie monitorovaného súboru. Na tento test je nutné nastaviť skupinu operácií *Write* na hodnotu *Log* alebo *No Log*. Dôvodom je zákaz premiestnenia či premenovania súboru s hodnotou *Deny*. Po nastavení skupiny *Write* na reakciu *Log* a po aktualizovaní zoznamu v driveru je všetko pripravené na analyzovanie obsahu. Po premiestení monitorovaného súboru nastanú dve udalosti. Prvou udalosťou je doplnenie konfiguračného súboru novým pravidlom, ktorý obsahuje aktualizované miesto alebo meno súboru. Druhá udalosť je vytvorenie nového záznamu v log súbore. Príklad záznamu ukazuje obrázok č. 9.

```
Time: 2011.4.30 11:38:55.986 Move/Rename Logged
from: \Device\HarddiskVolume1\Documents and Settings\All Users\Desktop\from.txt
to: \Device\HarddiskVolume1\Documents and Settings\All Users\Desktop\to.txt
```

Obrázok 9: Príklad vytvoreného záznamu pri premenovaní súboru v log súbore.

Posledný základný test slúži na otestovanie možnosti dočasného vypnutia pravidiel. V teste je na multimediálny súbor nastavená hodnota *Deny* na skupinu operácií *Read* i *Write*. Po aplikovaní bezpečnostnej politiky je každá práca so súborom zakázaná. Pomocou parametra *-echoconf* nájdeme v konfiguračnom súbore pravidlo, ktoré chceme dočasne vypnúť a príslušný index pravidla. Po nájdení spustíme užívateľskú aplikáciu s parametrom *-ruleoff n*, kde *n* je index pravidla a bezpečnostné pravidlo dočasne vypneme, čo umožní pracovať so súborom. Po práci môžeme pravidlo aktivovať pomocou parametra *-ruleson* či *-ruleon n*, kde *n* je už spomenutý index pravidla.

6.2 Pokročilé bezpečnostné testy

V časti pokročilé testy sú popísané a vyskúšané 3 potenciálne útoky, ktoré môžu slúžiť na otestovanie Personal DLP.

Pomocou prvého a druhého testu bolo vyskúšané oklamanie zákazu odstránenia súboru pri bezpečnostnej politike s reakciou *Deny* na skupinu operácií *Write*. Prvá technika spočívala v ukončení procesu *Explorer.exe*. Pri teste boli vykonané nasledujúce kroky:

- Spustenie príkazového riadku;
- Spustenie Task manageru;

- Cez Task manager bol ukončený proces *Explorer.exe*;
- V otvorenom príkazovom riadku bol zadaný príkaz *del subor.txt*.

Po vykonaní týchto krokov však v príkazovom riadku dostaneme oznámenie „Access is denied“, čo znamená úspešnosť ochranného systému Personal DLP.

Pri druhom teste bola použitá aplikácia *Unlocker*, ktorá je dostupná na webovej stránke [21]. Cieľom aplikácie je sprístupniť prácu so súbormi, s ktorými užívateľ kvôli nedostatku práv nemôže vykonať žiadané operácie. Po nainštalovaní aplikácie do systému sa objaví v rozbaľovacom menu u súborov možnosť používania programu *Unlocker*. Pri kliknutí na možnosť sú ponúknuté rôzne vykonateľné možnosti, z ktorých si vyberieme odstránenie súboru. Operácia sa však nie je vykonaná a dostaneme oznámenie o tom, že objekt nie je možné odstrániť. Vykonaný test môžeme pokladať ako úspešný.

V poslednom pokuse je simulované odcudzenie citlivých dát pomocou virtuálneho útoku, čo znamená útok cez internetu. Na simuláciu úniku bola používaná aplikácia *Poison Ivy 2.3.2*. Aplikácia je dostupná na [22]. Za účelom testovania bol vytvorený server, ktorý bol spustený na cieľovej stanici, z ktorého chce útočník ukradnúť dáta. Na počítači útočníka bola spustená aplikácia klienta, pomocou ktorého má možnosť útočník pripojiť sa na stanicu obete. Po úspešnom pripojení je útočník schopný stiahnuť ľubovoľný súbor z cieľovej stanice. Po nájdení súboru, ktorý je monitorovaný systémom Personal DLP sa pokúsime stiahnúť súbor na pevný disk útočníka. Bezpečnostné pravidlo používané v teste zakázalo skupinu operácií *Read* i *Write*. V nastavenom adresári, do ktorého aplikácia *Poison Ivy* si uložila odcudzené dáta, pribudol nový súbor s názvom cieľového súboru. Po otvorení však vidíme že súbor je prázdny. Dôvodom je, že systém Personal DLP zabránil útočníkovi odcudzenie súboru.

6.3 Možné vylepšenia a nedostatky systému

Táto kapitola sa zaoberá so známymi nedostatkami a chybami vytvoreného systému. Sú tiež popísané možnosti rozšírenia a reálneho uplatnenia Personal DLP. Medzi nedostatkami systému samozrejme nemôžeme zaradiť zlyhanie iných faktorov bezpečnosti, keď nespoľahlivý užívateľ získa administrátorské práva. V tomto prípade získa možnosť na vypnutie Personal DLP pomocou filter manageru, či na zmazanie obsahu konfiguračného súboru priamo, bez zadania hesla v užívateľskej aplikácii.

Za najvýznamnejší nedostatok systému môžeme považovať neschopnosť monitorovania kopírovacích operácií, ktorá už bola spomenutá v kapitole 6.1. Nemožnosť sledovania umožní duplikovanie sledovaných súborov bez toho aby novo vytvorené súbory boli tiež monitorované. Ako riešenie problému je možné nastaviť skupinu operácií *Read* i *Write* na hodnotu *Deny* a v prípade potreby práce so súborom dočasne vypnúť bezpečnostné pravidlo, ktoré bráni výkonu operácie.

Možnosti vylepšenia systému sú rozsiahle. Do budúcnosti sa rysujú dve konkrétne úpravy.

Prvým je vytvorenie crawlera (viď kapitolu 2.2). Crawler bude vytvorený v skriptovacom jazyku Python a bude mať za úlohu hľadanie textových súborov s konkrétnym obsahom. Hľadaný obsah bude definovaný pomocou regulárnych výrazov. Po nájdení súboru s citlivým obsahom crawler automaticky pridá bezpečnostné pravidlo do konfiguračného súboru s potrebnými parametrami, ktoré budú definované užívateľom.

Druhým plánovaným rozšírením bude aplikácia, ktorá umožní monitorovať tie užívateľské operácie, ktoré pomocou IRP správ nie sme schopní sledovať. Takéto operácie sú napríklad copy-paste obsahu alebo print screen obrazovky.

6.4 Testovanie spomalenia systému

V tejto časti práce je analyzované spomalenie systému pri behu aplikácie Personal DLP v závislosti od počtu monitorovaných súborov.

Na tento účel boli pomocou skriptu vygenerované textové súbory. Každý súbor mal meno *suborX.txt*, kde *X* mal hodnotu z množiny *m={0,49999}*. Obsah súboru tvoril príkaz príkazového riadku, ktorý pridáva do konfiguračného súboru samotný súbor. Napríklad obsah súboru *subor7.txt* bol *user.exe -add c:\dir\subor7.txt log log*. Dôvodom v značnej miere rovnakých mien súborov je úmyselné spomalenie porovnávacieho algoritmu reťazcov v jazyku C.

Na meranie času je používaná aplikácia *Timeit*, ktorá ako súčasť Windows Resource Kit je dostupná na [23]. Namerané výsledky nemusia byť úplne presné, kvôli nedostatku informácií o réžii otvárania samotného programu v čom sú otvárané súbory. Výsledky sú však postačujúce na zobrazenia rýchlosti Personal DLP a na zobrazenie spomalenie operačného systému pri behu implementovaného ovládača. Testy boli vykonané na virtuálnom počítači, v operačnom systéme Windows XP.

V tabuľke č. 4 sú kategorizované namerané výsledky, ktoré sú tiež znázornené v podobe grafu na obrázku č. 10. Testy sú rozdelené do štyroch skupín.

V prvej skupine je meraná otváracia doba súborov u ktorých nie je monitorovaný súbor a ani prípona súboru nepatrí do pozorovaných. V našom prípade to je každý súbor s inou príponou ako txt. Vo výsledkoch vidíme, že počet monitorovaných prvkov nemá vplyv na čas otvárania u tejto skupiny súborov. Za túto vlastnosť je zodpovedný zoznam, v ktorom sú uložené sledované prípony a tým umožní rýchle vyfiltrovanie súborov, ktoré majú nemonitorovanú príponu.

Druhú skupinu tvoria súbory, ktoré nie sú monitorované, ale na rozdiel od predchádzajúcej skupiny majú sledovanú príponu. Vo výsledkoch tabuľky vidíme, že čas otvárania je závislý na počtu monitorovaných súborov. Toto správanie sa bolo predpokladané, nakoľko pred otvorením súboru je potrebné prehľadať celý zoznam sledovaných súborov.

V tretej skupine sú testované súbory, ktoré sú sledované aplikáciou Personal DLP. Čas otvárania u tejto skupiny je vypočítaný pomocou aritmetického priemeru časov otvorenia prvého, posledného a stredného záznamu v zozname monitorovaných súborov. V nameraných hodnotách u tejto skupiny je možné vidieť závislosť na počte súborov sledovaných pomocou Personal DLP. Značnú rolu v zhoršení výsledkov hral posledný záznam v zozname a záznam uložený v strede zoznamu. Rast času v závislosti na počte monitorovaných súborov bol očakávaný, keďže prehľadanie zoznamu s vysokým počtom prvkom si žiada viacej procesorového času.

Poslednú skupinu tvorí čas otvárania cacheovaných súborov. Personal DLP uloží do zoznamu cache posledných desať otvorených monitorovaných súborov. Vo výsledkoch vidíme, že počet monitorovaných súborov nemá vplyv na rýchlosť otvárania. Dôvodom rýchleho otvárania týchto súborov bez ohľadu na množstvo uložených prvkov v zozname spočíva v testovaní cache pred ostatnými zoznamami.

Počet monitorovaných súborov	Súbor ani prípona nemonitorovaná	Súbor nesledovaná, prípona monitorovaná	Súbor monitorovaná, prvý pokus	Súbor monitorovaná, ďalší pokus
100	0.140s	0.089s	0.078s	0.078s
500	0.093s	0.095s	0.077s	0.078s
1000	0.125s	0.109s	0.067s	0.093s
5000	0.187s	0.109s	0.093s	0.046s
10000	0.125s	0.125s	0.124s	0.109s
50000	0.109s	0.156s	0.218s	0.125s

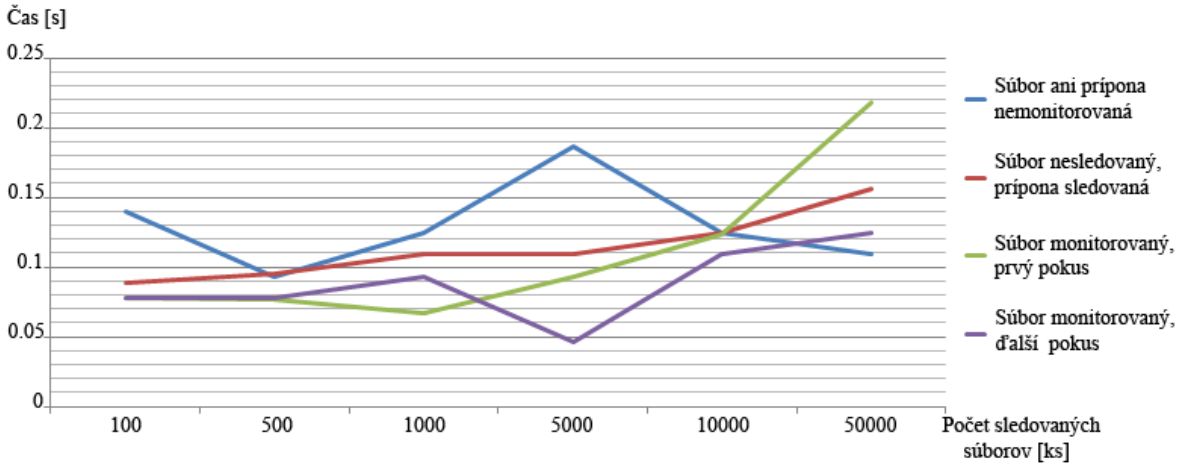
Tabuľka 4: Čas otvorenia súborov v závislosti na počte súborov.

Z nameraných výsledkov je možné posúdiť, že aplikácia Personal DLP ani pri pomerne veľkom počte monitorovaných súborov nespôsobí výrazné spomalenie operačného systému, čo bolo jedným hlavným cieľom pri navrhovaní aplikácie.

Pri testovaní spomalenia systému bol odmeraný aj čas potrebný na komunikáciu medzi užívateľskou aplikáciou a driverom. Výsledky merania sú v tabuľke č. 5. Celkový čas komunikácie je v značnej miere závislý na počte záznamov v konfiguračnom súbore. Dôvodom výrazného zhoršenia času je potreba overenia existencie každého súboru.

Počet súborov	100	500	1000	5000	10000	50000
Čas potrebný na komunikáciu	0.57s	1s	1.89s	8.22s	21.12s	3m 49.25s

Tabuľka 5: Potrebný čas na komunikáciu medzi driverom a user application.



Obrázok 10: Potrebný čas na otvorenia súborov v závislosti na počte monitorovaných súborov v podobe grafu.

7 Záver

Cieľom vytvorenej práce bolo predstaviť DLP systémov, spôsob ich činnosti a časti z ktorých sa skladajú. V práci je popísaný dôvod nevyhnutnosti používania týchto systémov u veľkých spoločností a výhody, ktoré môžu byť očakávané pri ich aplikovaní. Sú tiež spomenuté nevýhody a slabé miesta Data Loss Prevention systémov.

Druhým hlavným cieľom dokumentu bolo navrhnutie aplikácie, ktorá na princípe DLP systémov umožní čo najefektívnejší spôsob ochrany citlivých dát a duševného vlastníctva v operačnom systéme Windows. Sú popísané možnosti monitorovania a teoretické znalostí o ovládačoch a IRP správach, potrebné k riešeniu problému.

Za účelom splnenia cieľu bol navrhnutý a implementovaný bezpečnostný systém, ktorý sa skladá z dvoch hlavných častí. Prvá časť bola vytvorená po konzultácií s odborníkmi zo spoločnosti TrustPort a. s. Je to file system mini-filter driver implementovaný v jazyku C, ktorý na úrovni jadru operačného systému Windows filtruje IRP správy. Operačný systém používa spomínané IRP správy na komunikáciu medzi ovládačmi. Metóda sledovania nízkoúrovňových IRP štruktúr umožní efektívne monitorovanie práce s citlivými dátami. Efektivita metódy spočíva v monitorovaní činnosti so súbormi pod WinAPI[10] funkcií. Implementovaný driver umožní užívateľovi reagovať na vybrané operácie, ktoré môžu byť zakázané či zaznamenané podľa nastavenia bezpečnostných politík. Druhá časť systému je konzolová užívateľská aplikácia implementovaná v jazyku C++. Užívateľská aplikácia slúži ako komunikačný kanál medzi ovládačom a užívateľom, ktorý umožňuje spravovať a aplikovať bezpečnostné politiky a zobraziť zaznamenané bezpečnostné udalosti.

Jedným z hlavných nevýhod DLP systémov, kvôli ich komplexnosti, je náročnosť behu na systém. Preto medzi ciele pri návrhu patrilo aj dosiahnutie adekvátnej rýchlosti monitorovania operácií so súbormi, aby vytvorený systém bol použiteľný bez výrazného spomalenia operačného systému. Tento cieľ bol úspešne splnený pomocou viacúrovňového filtrovania IRP správ. Ako dôkaz na splnenie cieľu bol vytvorený systém testovaný pri sledovaní až päťdesiattisíc súborov, pri ktorom nedošlo k významnému spomaleniu operačného systému.

Vytvorený systém môže slúžiť ako základ bezpečnostného systému. V budúcnosti plánujem program rozšíriť, aby umožnil automatické identifikovanie citlivých dát či monitorovanie užívateľských operácií ako Print Screen otvoreného citlivého obsahu, ktoré nie je možné sledovať pomocou IRP správ. Ďalším cieľom je vytvorenie grafického užívateľského rozhrania, čo by umožnilo užívateľovi používať konzolovú alebo grafickú verziu na komunikáciu s ovládačom a na spravovanie pomocných súborov. Konečným cieľom tohto projektu je vytvoriť komponentu personálneho DLP integrovateľnú do existujúceho antivírového riešenia.

[10] **WinAPI** je sada rozhraní, od firmy Microsoft, pre programovanie aplikácií [25].

Literatura

[1] Computer virus. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 29 September 2001, last modified on 4 April 2011 [cit. 2011-05-09]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Computer_virus>.

[2] MOGUL, Rich. Understanding and Selecting a Data Loss Prevention Solution. In *Understanding and Selecting a Data Loss Prevention Solution* [online]. USA : The SANS Institute, 2007 [cit. 2011-05-09]. Dostupné z WWW: <http://www.sans.org/reading_room/dlp/87.pdf>.

[3] Data Leak Prevention. In *Data Leak Prevention* [online]. USA : Isaca, 2010 [cit. 2011-05-09]. Dostupné z WWW: <http://www.isaca.org/Knowledge-Center/Research/Documents/DLP-WP-14Sept2010-Research.pdf>.

[4] PRINCE, Kevin. *Network Security Edge* [online]. 2010-01-29 [cit. 2011-05-09]. Top 10 Information Security Threats of 2010. Dostupné z WWW: <http://www.networksecurityedge.com/content/top-10-information-security-threats-2010>.

[5] COLWILL, Carl. Human factors in information security: The insider threat : Who can you trust these days?. *Information Security Technical Report* [online]. 11-2009, vol. 14, no. 4, [cit. 2011-05-09]. Dostupný z WWW: <http://www.infosec.co.uk/files/istr_article_on_risk.pdf>. ISSN 1363-4127.

[6] Deployment and Installation Guide. In *Deployment and Installation Guide Websense Data Security* [online]. USA : Websense Inc., 2010 [cit. 2011-05-09]. Dostupné z WWW: <http://ebookbrowse.com/data-security-deployment-and-installation-guide-7-5-3-pdf-d42257872>.

[7] RYAN, Pearse; HARBISON, Andrew. Data leakage and data compromise : Causes and preventative steps. In *Data leakage and data compromise Causes and preventative steps* [online]. England : Grant Thornton, 2009 [cit. 2011-05-09]. Dostupné z WWW: <http://www.grantthornton.ie/db/Attachments/Publications/Forensic_&_inve/Grant%20Thornton%20Data%20leakage%20and%20data%20compromise%20September%202009.pdf>.

[8] Phishing. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13. november 2005, last modified on 6. február 2011 [cit. 2011-05-09]. Dostupné z WWW: <http://sk.wikipedia.org/wiki/Phishing>.

[9] Trójsky kôň (informatika). In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 18. august 2005 , last modified on 24. marec 2011 [cit. 2011-05-09]. Dostupné z WWW: <http://sk.wikipedia.org/wiki/Trôjsky_kôň_(informatika)>.

[10] *Federal Information Security Management Act* [online]. 2002 [cit. 2011-05-09]. INFORMATION SECURITY. Dostupné z WWW:

<http://www.marcorsyscom.usmc.mil/sites/pmia%20documents/documents/Federal%20Information%20Security%20Management%20Act%20%28FISMA%29.htm>.

[11] *ISO 27001* [online]. 2005 [cit. 2011-05-09]. INFORMATION SECURITY MANAGEMENT DEFINITIONS. Dostupné z WWW: <http://www.praxiom.com/iso-27001-definitions.htm>.

[12] Hooking. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 20 October 2005 , last modified on 3 January 2011 [cit. 2011-05-09]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Hooking>.

[13] RUSSINOVICH, Mark; SOLOMON, David. *Windows Internals : Including Windows Server 2008 and Windows Vista*. 5th edition. Redmond (Washington) : Microsoft Press, 2009. 1232 s. ISBN 9780735625303.

[14] ONEY, Walter. *Programming the Microsoft Windows Driver Model*. 2nd edition. Redmond (Washington) : Microsoft Press, 2002. 880 s. ISBN 0735618038.

[15] *MSDN Library* [online]. 2011-03-05 [cit. 2011-05-09]. IRP Major Function Codes. Dostupné z WWW: <http://msdn.microsoft.com/en-us/library/ff550710%28v=VS.85%29.aspx>.

[16] *Microsoft Technet* [online]. 2003-03-28 [cit. 2011-05-09]. What Is a Device Driver?. Dostupné z WWW: <http://technet.microsoft.com/en-us/library/cc776246%28WS.10%29.aspx>.

[17] *Windows Hardware Developer Center* [online]. 2003 [cit. 2011-05-09]. File System Filter Drivers. Dostupné z WWW: <http://msdn.microsoft.com/en-us/windows/hardware/gg462968>.

[18] Filter Driver Development Guide. In *Filter Driver Development Guide* [online]. Redmond (Washington) : Microsoft Corporation, 2004 [cit. 2011-05-09]. Dostupné z WWW: <http://download.microsoft.com/download/e/b/a/eba1050f-a31d-436b-9281-92cdfeae4b45/filterdriverdeveloperguide.doc>.

[19] *Microsoft Help and Support* [online]. 2006-11-21 [cit. 2011-05-09]. Understanding Device Names and Symbolic Links. Dostupné z WWW: <http://support.microsoft.com/kb/235128>.

[20] *Microsoft Download Center* [online]. 2010-02-26 [cit. 2011-05-09]. Windows Driver Kit Version 7.1.0 . Dostupné z WWW: <http://www.microsoft.com/downloads/en/details.aspx?displaylang=en&FamilyID=36a2630f-5d56-43b5-b996-7633f2ec14ff>.

[21] *Unlocker 1.9.1* [online]. 2011 [cit. 2011-05-09]. Dostupné z WWW: <http://www.emptyloop.com/unlocker/>.

[22] *Poison Ivy* [online]. 2007 [cit. 2011-05-09]. Remote Administration Tool. Dostupné z WWW: <http://www.poisonivy-rat.com/index.php?link=download>.

[23] *Microsoft Download Center* [online]. 2003-04-28 [cit. 2011-05-09]. Windows Server 2003 Resource Kit Tools. Dostupné z WWW: <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9d467a69-57ff-4ae7-96ee-b18c4790cffd&DisplayLang=en>.

[24] KUROSE, James F.; ROSS, Keith W. *Computer Networking : A Top-Down Approach Featuring the Internet*. 3rd edition. Boston (Massachusetts) : Addison-Wesley, 2004. 848 s. ISBN 9780321227355.

[25] Windows API. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 6 September 2002, last modified on 20 March 2011 [cit. 2011-05-14]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Windows_API>.

Seznam příloh

Příloha 1. Obsah CD

- Manual/ - manuál systému Personal DLP
- Src/ - zdrojové kódy
    - filter/ - zdrojové kódy vytvoreného mini-filter driveru
    - user/ - zdrojové kódy vytvorenej užívateľskej aplikácie
- Readme.txt - Readme vytvoreného systému Personal DLP